Programação em MaxScript para Iniciantes

por: Tito Petri. *www.titopetri.com.br*

1. Apresentação do Material

Olá eu sou o Tito Petri e seja bem-vindo à apostila de MaxScript para Iniciantes**.**

Nesse documento vou te mostrar tudo o que conheço sobre Programação de Ferramentas, Plugins e Rotinas para o Autodesk 3D Studio Max, utilizando a Linguagem MaxScript.

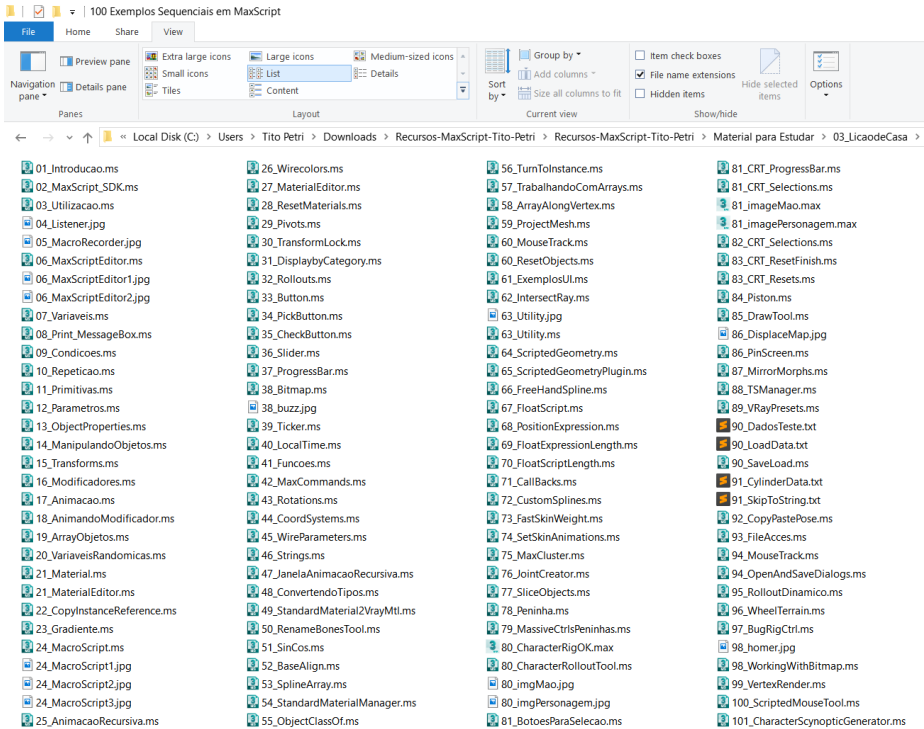
Espero que o material seja útil para seus estudos e te ajude a alcançar seus objetivos.

Um Grande Abraço!

Tito Petri

2. Download dos Arquivos

Antes de começar a ler este material, eu sugiro que você faça o **download** do meu pacote de Recursos de MaxScript.

Trata-se de um pacote de arquivos que incluem minhas Ferramentas Criadas para o meu dia a dia de trabalho no 3DSMAX.

Neste pacote de arquivos você encontra todos os exemplos e códigos que estudaremos neste livro e também um diretório com mais de **100 Exemplos de Scripts** ordenados por **nível de dificuldade**, para o aluno explorar um a um e aprender ainda mais.

Curso Online em Vídeo

Você também pode aprender MaxScript assistindo às minhas videoaulas sobre o assunto no **Curso Online de MaxScript** com mais de 80 vídeos, somando 5 horas de treinamento.

3. O Que é MaxScript?

- MaxScript é uma linguagem de programação orientada a objeto mais simples e fácil que o C++.

- O MaxScript foi um plugin introduzido no 3D Studio Max R2, e mais tarde incorporado ao 3D Studio Max.

- MaxScript não é uma linguagem para se escrever plugins. Plugins são criados em C++ usando-se o Max SDK (Software Development Kit), que vem com a licença do 3D Studio Max.

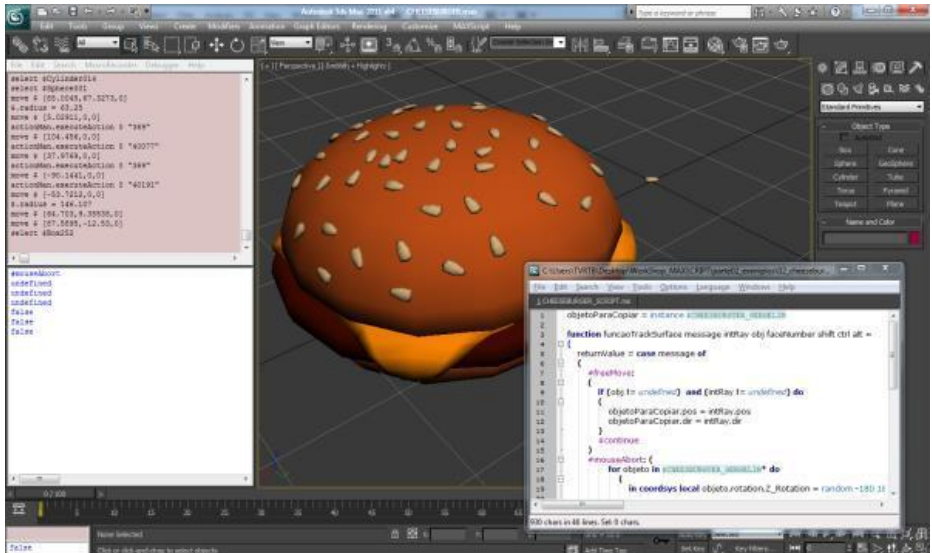
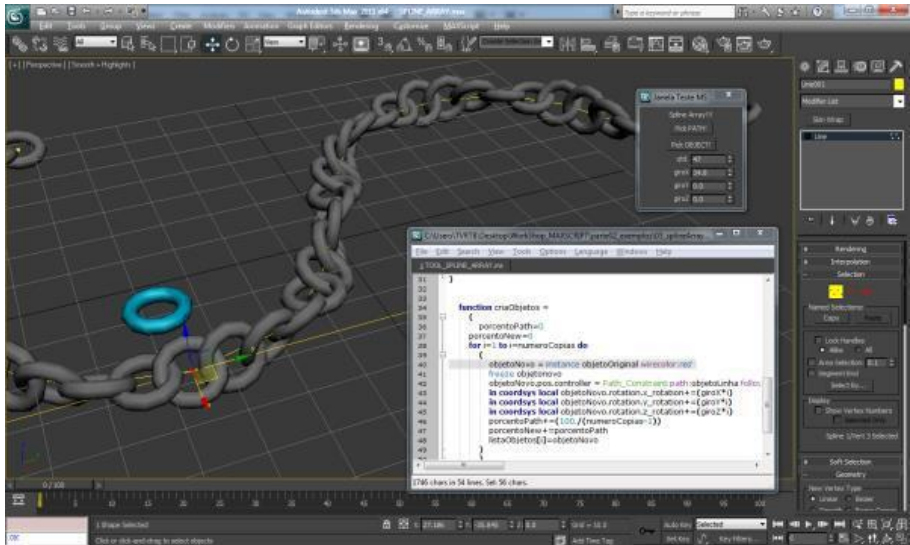
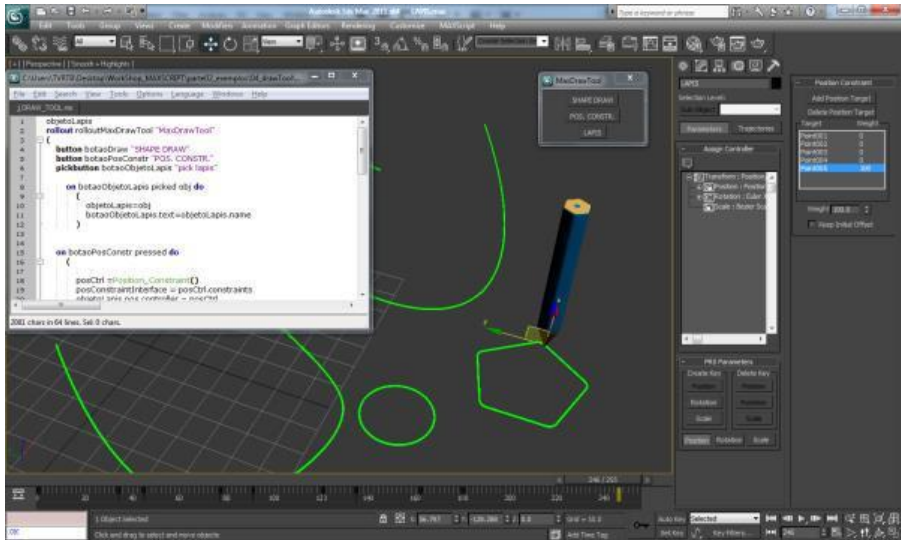
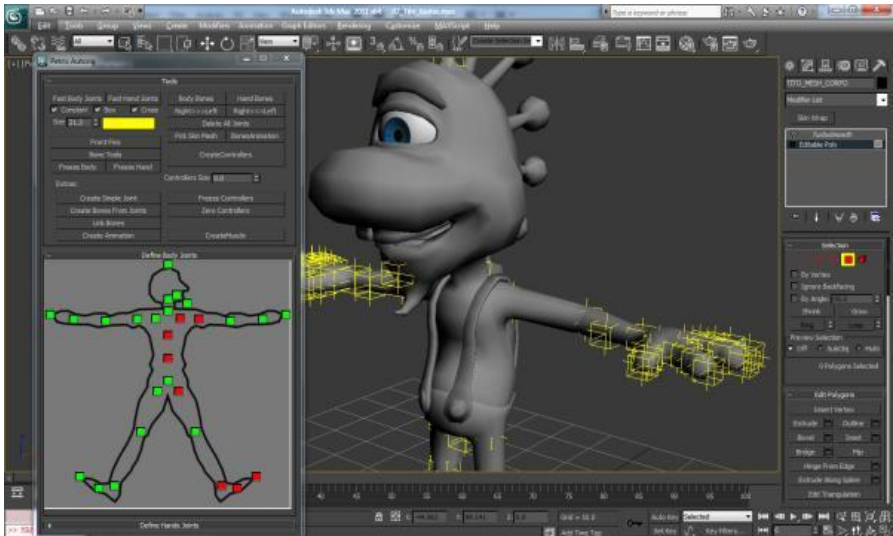
- MaxScript e Plugins são diferentes. Os usuários não teriam a capacidade de identificar a diferença, os dois adicionam funcionalidades ao 3DSMax.

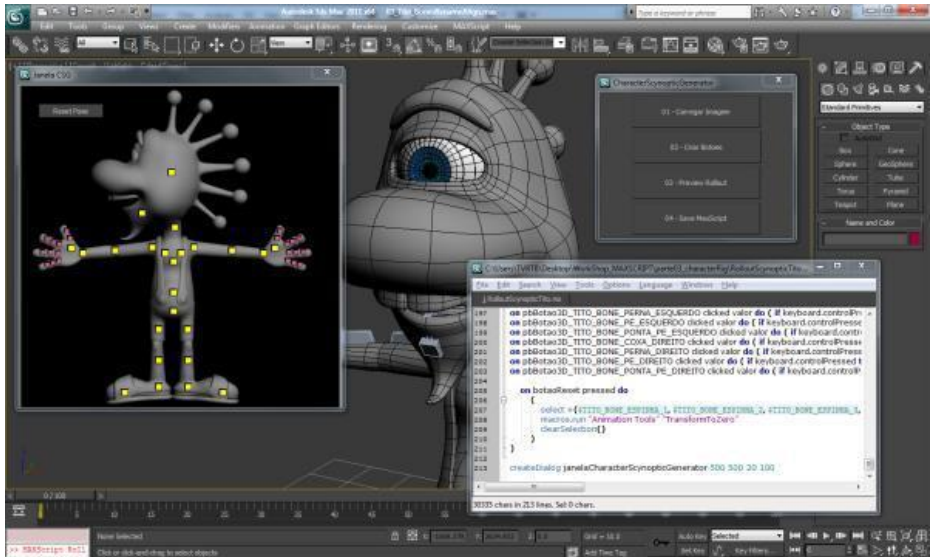
- Scripts são mais limitados, e têm um processamento mais demorado, porém são mais fáceis de se escrever.

Utilizações do MaxScript:

- Agilidade na execução de tarefas repetitivas.
- Criar novas ferramentas e funções no 3DSMAX.
- Aprimorar rigs e setups.
- Aplicar lógica e matemática.

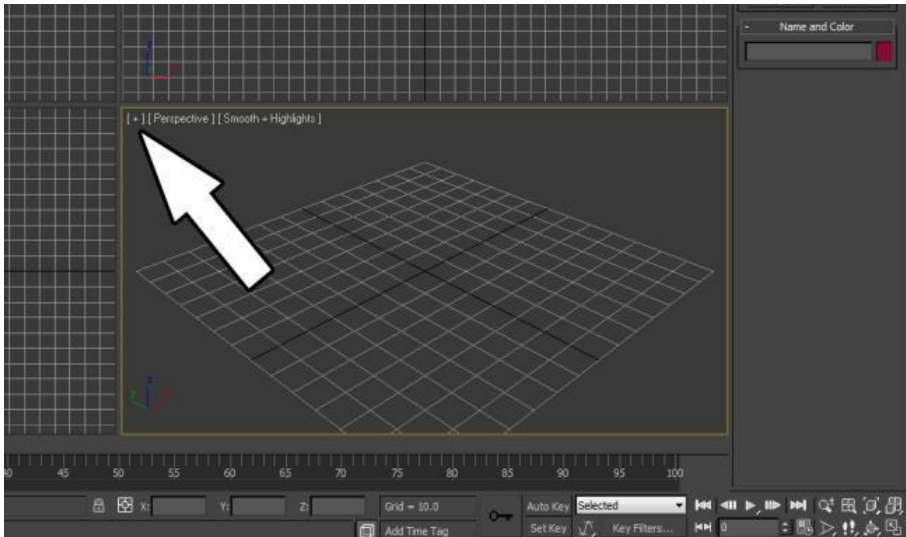
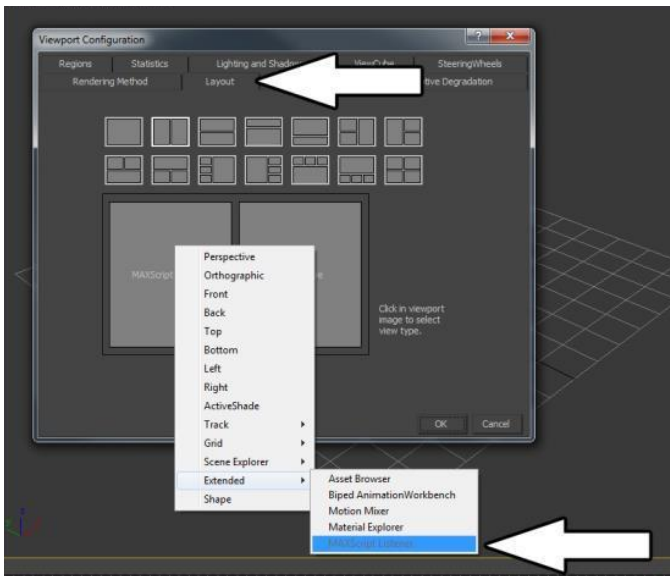
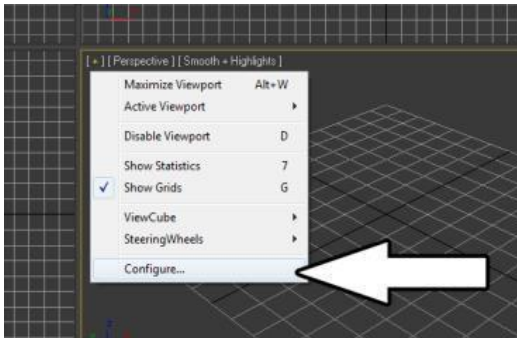
Observe algumas situações interessantes onde foi utilizado o MaxScript:

MaxScript utilizado para espalhar um determinado objeto sobre outra superfície.

MaxScript para replicar um objeto ao longo de uma linha.

MaxScript para animar o desenho de linhas.

Ferramenta criada em MaxScript para configurar as juntas do esqueleto no personagem.

Ferramenta para criar automaticamente ossos para o esqueleto do personagem baseado na criação das juntas.

Criação de Janelas de Interface para controlar o *rigging* do personagem.

Se quiser assistir ao funcionamento destas ferramentas, visite a [*playlist* do meu canal do Youtube - Demos de MaxScript no Youtube]().

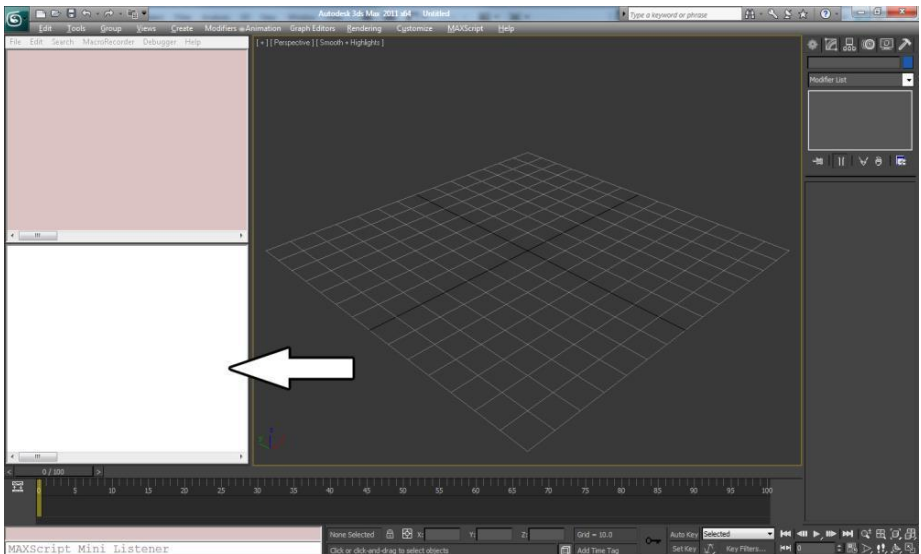
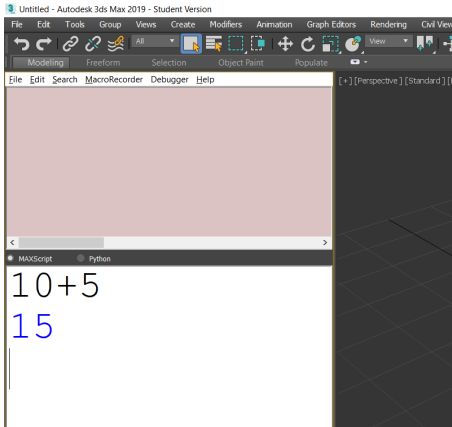
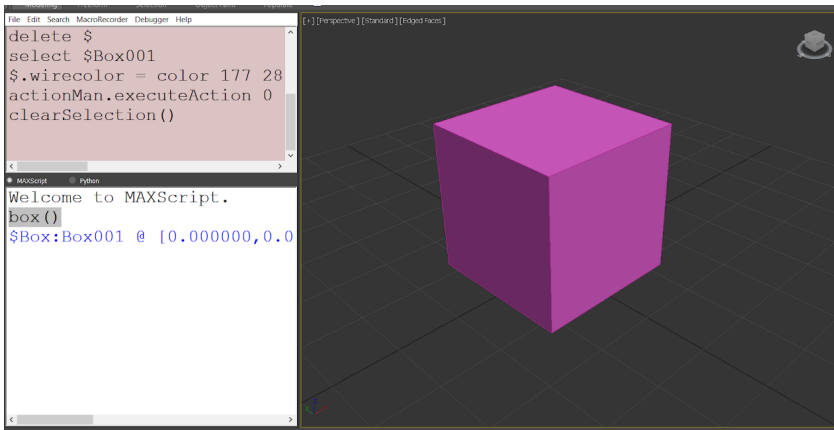
4. Primeiros Passos no MaxScript

Antes de começarmos a examinar e aprender os códigos, vamos primeiro preparar a interface do 3DSMAX para facilitar o trabalho, enquanto estivermos escrevendo e executando os códigos.

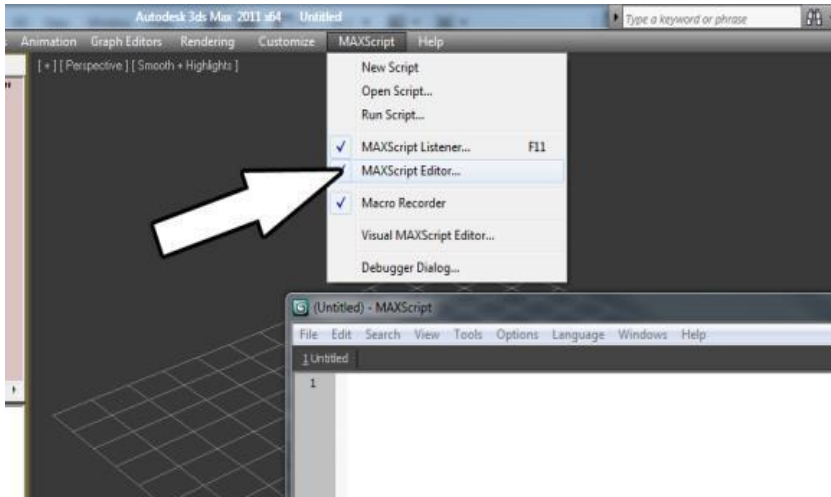
Ajustando a Interface do 3DSMAX

Clique em **[+]** e, depois, em **Configure.**

Em **Layout**, ajuste uma janela para **Perspective**

e a outra em **Extended** > **MaxSriptListener.**

MaxScript Listener

O Listener é o Console de Depuração, ou Debug. Ele que vai te dizer se um comando foi executado, qual foi o resultado de uma determinada operação ou se algo que você está tentando fazer está incoerente ou dando erros. Fique sempre atento a todas as mensagens que aparecerão no Listener, ele é o “Ouvidor” da cena, ele que vai te dizer tudo o que está acontecendo por trás da execução do programa.

Digite **10+5** (na parte branca da janela) e aperte **ENTER** (do teclado numérico!) para executar o comando.

Agora digite **Box()** e aperte o **ENTER do teclado numérico.**

Parabéns! Você executou o seu primeiro comando em MaxScript, para criar uma primitiva de um cubo, na cena do 3D Studio Max.

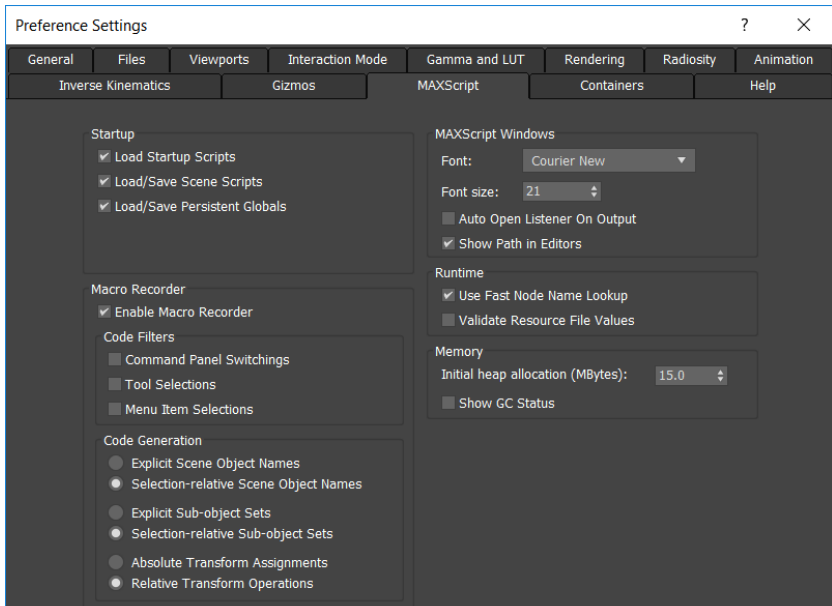
MaxScript Editor

O MaxScript Editor é uma espécie de “Bloco de Notas” para você escrever os seus MaxScripts.

Além dos recursos de um editor de texto comum, ele tem diversos recursos como colorir o código pela sintaxe do MaxScript, Editor Visual de Layout, entre outros recursos.

Atalhos úteis do MaxScript Editor:

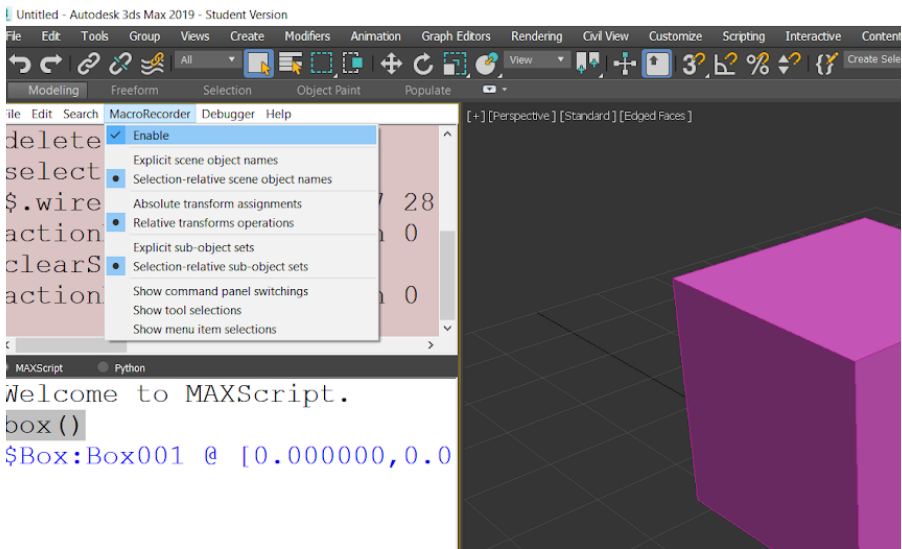
- Segurando o **CTRL** e rolando o **Scroll do mouse** você aumenta o tamanho da fonte.
- **CTRL+E** (Evaluate All) - Executa todo o Script.

- **SHIFT+ENTER** (ou ENTER do teclado numérico) - Executa apenas a linha selecionada.
- **CTRL+Q** - Comenta a linha atual.

Existe também um menu para configurar as preferências do MaxScript Editor. Para acessá-lo, vá em **File** > **Preferences** > **MaxScript**

Neste menu você pode configurar diversas opções do MaxScript e do Editor. Uma opção útil aqui será alterar o tamanho da fonte (indispensável quando falta o *scroll* do mouse, no caso de estar usando uma caneta ou tablet).

Macro Recorder

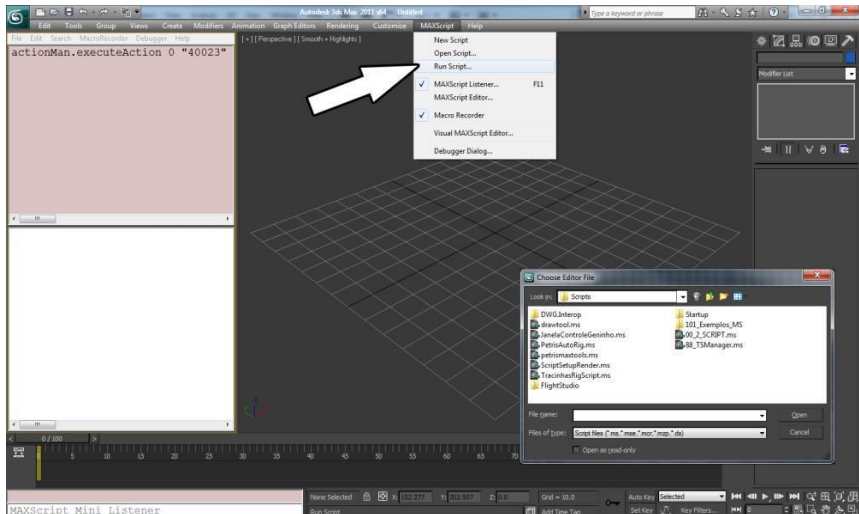
O **Macro Recorder** é a ferramenta que captura muitas das ações executadas pelo usuário e gera os comandos de MaxScript que correspondem a essas ações.

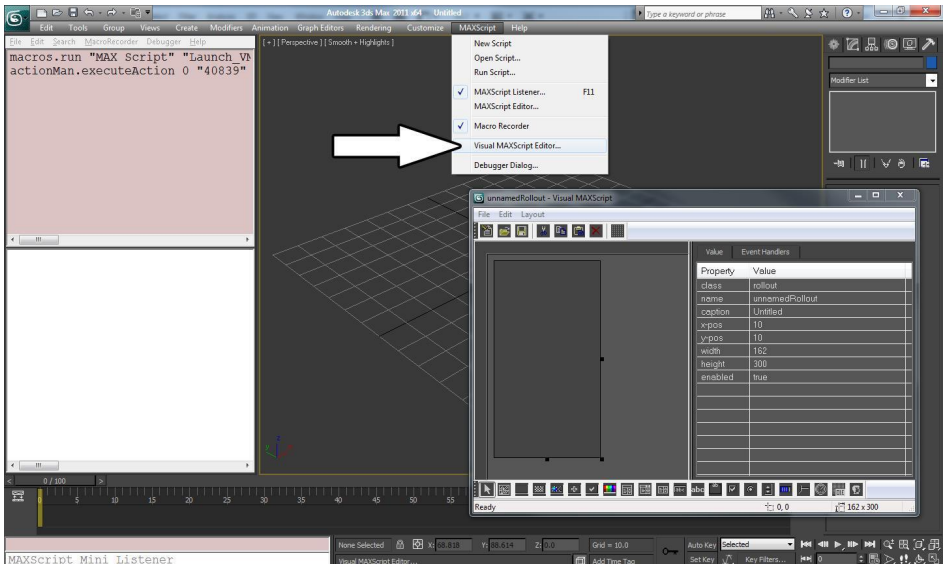
Habilite o macroRecorder, acessando as opções no Listener em **MacroRecorder** > **Enable.**

Agora o MacroRecorder estará mostrando em forma de macros todas suas ações no 3DSMAX. Experimente criar ou mover um objeto.

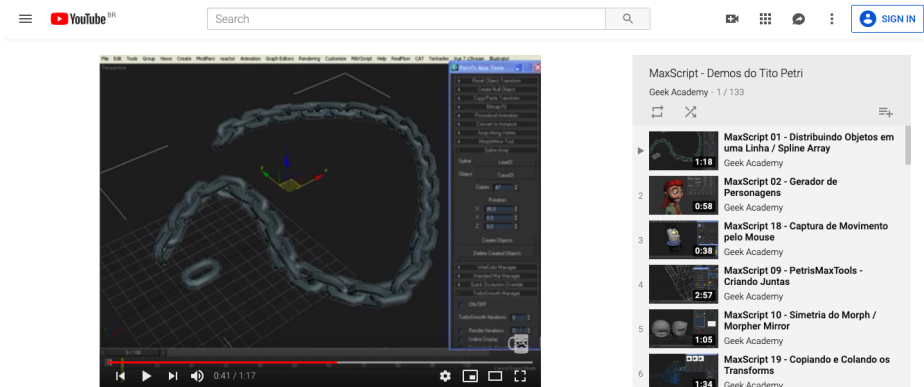
Dica: Para limpar o macroRecorder ou o Listener utilize atalho **CTRL+D.**

Executando um Arquivo MaxScript

Os scripts podem ter a extensão .ms ou .mse ou .mcr.

Para executá-los vá em **MAXScript** > **RunScript.**

Visual MaxScript Editor

Para criar rollouts mais complexos, existe um editor visual para criar os botões e fazer o layout da janela. Uma boa maneira de conhecer os recursos da UI.

Demos no YouTube

A esta altura, você já deve estar interessado em entender um pouco mais sobre o funcionamento do MaxScript, então eu sugiro que dê uma espiada nesta Playlist do meu canal do YouTube:

Ali, eu demonstro dezenas de ferramentas que criei utilizando o MaxScript, ao longo dos meus anos de trabalho com o 3DSMAX.

Lembre-se que a maioria dos Scripts que escrevi encontra-se dentro do pacote de recursos de MaxScript que você baixou, no início da apostila.

5. Lógica e Algoritmos com MaxScript

Nesta seção vamos conhecer um pouco sobre os primeiros passos na programação em MaxScript, como a criação de variáveis, estruturas de condição e repetição.

Se você não conhece absolutamente nada sobre programação, e nem sabe o que são esses termos que acabei de utilizar, eu sugiro que assista esta pequena lista de vídeos no meu Canal do YouTube. Nela explico em 30 minutos as primeiras idéias e conceitos sobre Lógica de Programação para Iniciantes.

Variáveis

Exemplo em MaxScript:

```
-- Alguns tipos de variáveis no MaxScript:

-- Textos ou strings.
palavraA = "WorkShop MaxScript"
palavraB = " - “Cursos Online do Felpudo"
titulo = palavraA+palavraB

-- Números inteiros (#integer).
numeroInteiro = 100
resultado = 50-30
resultadoMultiplicacao = resultado*10

-- Números decimais (#float).
numeroDecimal = 2.5

-- Booleans (#bool).
```

```
variavelBooleana = true

-- Vetor de 3 pontos XYZ (posições ou escalas) (#vec3f).
posicaoObjeto = [10,35,0]

-- Nodes ou objetos (#node).
objetoNovo = Teapot()
caixa = Box()
bolinha = Sphere radius:10 segs:12 pos:[10,0,0]
objetoIntersect = caixa-bolinha

-- Cores (#color).
corMaterialVermelho = color 255 50 50
corObjeto = red

-- Sorteio aleatório.
-- (Variável recebe aleatoriamente um número entre -10 e 10.)
numeroAleatorio = random -10 10

$.wirecolor = random black white
-- (Variável recebe aleatoriamente uma cor.)
```

Exibindo Alertas e Mensagens

Para depurar nossas aplicações, podemos usar alguns comandos para exibir mensagens tanto no console do Listener, quanto dentro do programa, através de janelas de diálogo.

Veja os comandos abaixo, e execute no seu Listener para conhecê-los.

Print

Exibe uma mensagem ou um valor no Listener.

Exemplo em MaxScript:

```
print ("Bem-vindos ao MaxScript!!!")
```

Podemos exibir mensagens, ou o conteúdo de variáveis, acessando seu valor através da **aspas simples**. Execute o comando **print** e verá uma mensagem sendo exibida no console do Listener.

Lembre-se que para juntar Strings (concatenar Strings), você pode acessar o conteúdo das variáveis utilizando as aspas simples.

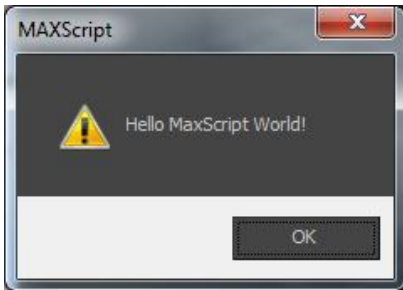
Exemplo em MaxScript:

```
palavraA = “Max”
palavraB = “Script”
print ("Welcome to "+'palavraA'+'palavraB')
```

MessageBox

Os MessageBox servem para exibir *pop-ups de* mensagens de alerta na tela do 3DSMax. Deve-se passar um String como argumento para utilizar este método.

Exemplo em MaxScript:

```
messagebox "Hello MaxScript World!"
```

Exemplo em MaxScript:

```
print ("Bem-vindos ao MaxScript!!!")

print corObjeto
print bolinha
print bolinha.name

messageBox ("O nome da bolinha é: " + (bolinha.name as String))
```

QueryBox

O QueryBox é o tipo de janela que retorna *true* ou *false*, de acordo com a resposta que o usuário dá à caixa de diálogo.

Exemplo em MaxScript:

```
if queryBox "Aviso MS!!!" then Box() else Sphere()
```

6. Controlando as Cenas do 3DSMAX com MaxScript

Nesta seção vamos começar a entender como se cria e configura os objetos na cena do 3D Studio Max.

Criação de Primitivas

Agora que já sabemos executar os scripts, vamos começar a manipular alguns objetos na *viewport* do 3DSMAX.

Observe no exemplo adiante como ocorre a **criação de uma primitiva**, e como podemos atribuí-la (ou guardá-la) em uma variável.

Exemplo em MaxScript:

```
-- Criando uma Primitiva do tipo Box.
Box()

-- Criando um objeto e armazenando em uma variável.
objetoA = Sphere()

-- Criando um objeto com parâmetros iniciais.
objetoB = Cone radius1:10 radius2:30

-- Manipulando a posição de um objeto através da sua variável.
objetoA.pos = [50,0,0]
objetoB.pos = [-50,0,0]

-- Configurando a posição de um objeto na hora da criação.
objetoC = Cylinder radius:20 height:50 pos:[50,50,50]
```

Parâmetros

Também podemos acessar os parâmetros, ou propriedades do objeto, através do ponto (.) ele indica que nos referimos a algum atributo deste objeto. Observe:

Exemplo em MaxScript:

```
-- Criando um objeto e atribuindo a ele uma variável.
novoObjeto = Sphere()

-- Acessando os parâmetros do objeto através da variável.
novoObjeto.radius = 10
novoObjeto.pos.z = 20
novoObjeto.segs = 64
novoObjeto.hemisphere = 0.5
novoObjeto.slice = on
novoObjeto.sliceFrom = 45
novoObjeto.sliceTo = -45
novoObjeto.pos = [0,0,10]
```

* Dica Importante!

Observe, adiante, o comando **showProperties.** Ele vai exibir no Listener todas as propriedades que existem em um determinado objeto, ou classe. Isto será muito útil para você desvendar quais são os comandos necessários para acessar os atributos dos objetos.

Propriedades dos Objetos

No exemplo a seguir, vamos criar uma primitiva Teapot e explorar as suas propriedades:

Exemplo em MaxScript:

```
-- Criamos um objeto do tipo Teapot e armazenamos em uma variável.
objeto = Teapot()

-- O comando ShowProperties exibe no listener todos os parâmetros do objeto.
ShowProperties objeto
ShowProperties(Bend())
ShowProperties(Cellular())

-- Além das propriedades, os objetos possuem um monte de outras propriedades e parâmetros.
objeto.pos.z = 15
objeto.scale = [1,1,3]

-- Opções que estão em Object Properties (botão direito).
objeto.showFrozenInGray = on
objeto.backfacecull = on
objeto.xray = on
```

Manipulando os Transforms

Analise cada um dos exemplos a seguir, de como podemos configurar os Transformas das geometrias (posição, rotação e escala X, Y e Z), com o MaxScript.

Repare que podemos guardar o valor da rotação (*eulerangles*) em uma variável para atribuir ao *transform* de rotação do objeto, mais adiante no código.

Exemplo em MaxScript:

```
-- Criando dois objetos.
bolinha = Sphere()
cubinho = Box()

-- Criando um objeto e definindo Bule como nome.
chaleira = Teapot name:"Bule"

-- Manipulando a posição da esfera.
bolinha.pos = [50,0,0]

-- Criando um objeto eulerangles.
angulo = eulerangles 0 0 45

-- Afetando a rotação do cubo.
cubinho.rotation = angulo

-- Pode-se acessar os objetos existentes na cena pelo $+ o nome do
objeto. Exemplo:
$Bule.radius = 10
$Bule.pos.x = -50
$Bule.scale.z = 5
```

Repare também que podemos acessar os objetos existentes na cena aberta através do Operador **$**.

Mais um exemplo de como podemos manipular a **posição**, **rotação** e **escala** dos objetos.

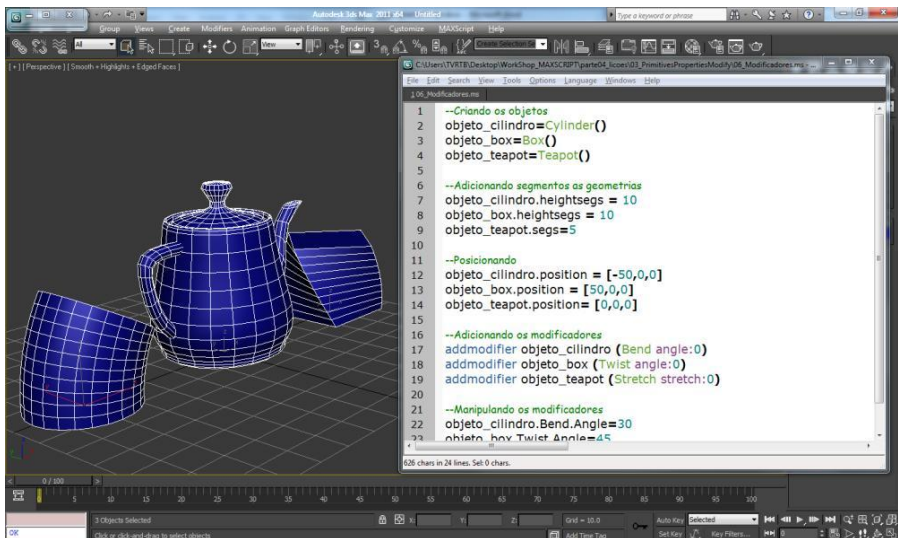
Exemplo em MaxScript:

```
-- Criando um objeto.
objeto = Box()

-- Guardando valores de posição, rotação e escala nas variáveis.
posicao = [0,50,0]
angulos = eulerAngles 0 0 45
escala = [2,2,2]

-- Aplicando as transformações ao objeto.
objeto.scale = escala
objeto.rotation = angulos
objeto.pos = posicao

-- Também podemos aplicar a transformação em um único eixo:
objeto.pos.z = 50
objeto.scale.y = 0.5
```

7. Modificadores, Materiais e Animação com MaxScript

Nesta seção vamos aprender como aplicar *Modifiers* (modificadores), criar e configurar os Materiais Básicos (*StandardMaterial*), como acessar o Editor de Materiais por MaxScript e como criar *keyframes* de animação por código.

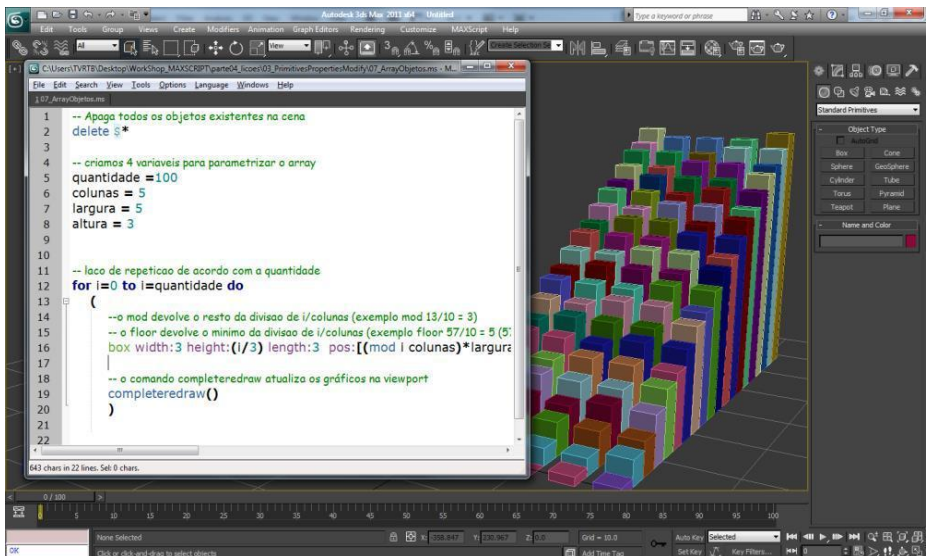
Modificadores

Agora vamos entender um pouco sobre como aplicar e configurar os **Modifiers** do 3DSMAX nos objetos através do código.

Exemplo em MaxScript:

```
-- Criando os objetos.
objeto_cilindro = Cylinder()
objeto_box = Box()
objeto_teapot = Teapot()

-- Adicionando segmentos às geometrias.
objeto_cilindro.heightsegs = 10
objeto_box.heightsegs = 10
objeto_teapot.segs = 5

-- Posicionando os objetos na cena.
objeto_cilindro.position = [-50,0,0]
objeto_box.position = [50,0,0]
objeto_teapot.position = [0,0,0]

-- Adicionando os Modificadores.
addmodifier objeto_cilindro (Bend angle:0)
addmodifier objeto_box (Twist angle:0)
addmodifier objeto_teapot (Stretch stretch:0)

-- Manipulando os Parâmetros dos Modificadores.
objeto_cilindro.Bend.Angle = 30
objeto_box.Twist.Angle = 45
objeto_teapot.Stretch.Stretch = .3
```

Array de Objetos

Podemos, também, utilizar estruturas de repetição para criar matrizes de objetos, ao invés de criá-los e configurá-los um a um.

Tentarei utilizar um exemplo bem simplificado aqui, mas saiba que quanto mais você conhecer de MaxScript e Matemática, mais vai ter controle sobre a criação ou a animação de **massas de objetos** (cenas com centenas, ou milhares de objetos, impraticáveis de serem manipulados um por um).

Exemplo em MaxScript:

```
-- Apagando todos os objetos existentes na cena.
delete $*
```

```
-- Criando 4 variáveis para parametrizar o array.
quantidade = 100
colunas = 5
largura = 5
altura = 3

-- Laço de repetição de acordo com a quantidade.
for i=0 to i=quantidade do
(
--O mod devolve o resto da divisão de i/colunas (exemplo mod 13/10 = 3).
-- O floor devolve o número inteiro da divisão de i/colunas.
-- Eexemplo floor 57/10 = 5.

sphere radius:1 pos:[(mod i colunas)*largura,(floor (i/colunas))*altura,0]

-- Atualiza os gráficos na viewport.
completeredraw()
)
```

Material Básico - Standard

Agora, vamos criar e configurar alguns materiais.

Exemplo em MaxScript:

```
-- Cria um objeto.
objeto = Teapot segs:24

-- Cria uma nova variável do tipo material.
novoMaterial = StandardMaterial()

-- Altera as propriedades do material.
novoMaterial.diffuseColor
novoMaterial.Diffuse = color 90 90 250
novoMaterial.specularLevel = 125
novoMaterial.glossiness = 50

-- Aplica o material ao objeto.
objeto.material = novoMaterial

-- Atualiza a viewport.
CompleteRedraw()
```

Editor de Materiais

Também podemos acessar e manipular as propriedades de um material dentro do Editor de Materiais.

Exemplo em MaxScript:

```
-- Cria um objeto.
bolinha = Sphere segs:64

-- No editor de materiais, podemos acessar os slots e suas propriedades
pelo seu número de 1 a 24.
meditMaterials[3].name = "Material da Bolinha"
meditMaterials[3].Diffuse = yellow
meditMaterials[3].specularLevel = 25
meditMaterials[3].glossiness = 35
meditMaterials[3].opacity = 50
meditMaterials[3].twoSided = on

-- Aplica o material do slot 3 no objeto bolinha.
bolinha.material = meditmaterials[3]

-- Atualiza a viewport.
CompleteRedraw()

-- Seleciona entre os modos COMPACT ou SLATE.
MatEditor.mode = #basic
MatEditor.mode = #advanced

-- Abre/fecha o Editor de Materiais.
MatEditor.Open()
MatEditor.Close()

-- Cria um novo material em cada slot do editor.
for i = 1 to i = 24 do
(
	MeditMaterials[i] = StandardMaterial ()
)
```

Animando Objetos pelo Script

Podemos também usar o MaxScript para criar *keyframes* de animação na nossa cena.

Exemplo em MaxScript:

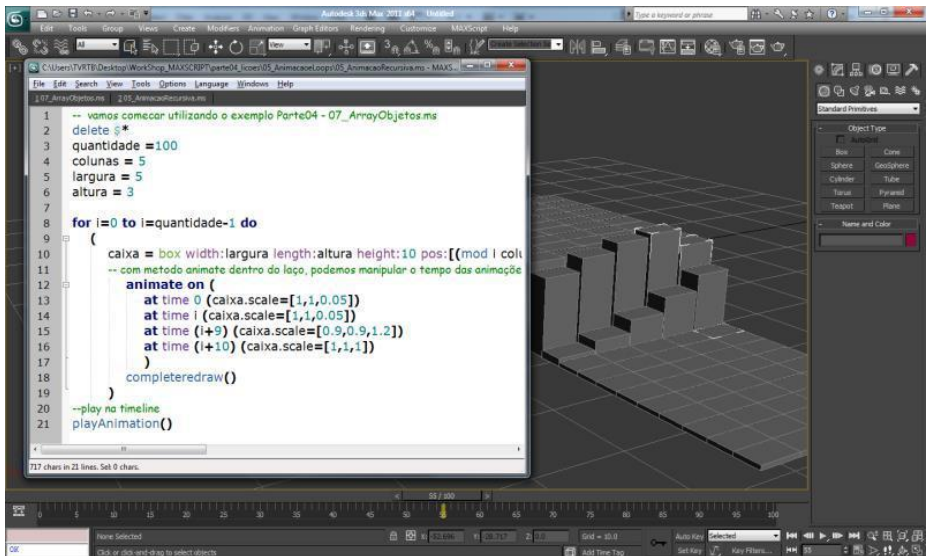
```
objeto = Box()

-- Método de Animação de Objetos:

animate on
(
	at time 0 (objeto.pos = [0,0,0] ; objeto.scale = [1,1,1])
	at time 10 (objeto.pos = [0,0,50] ; objeto.scale = [0.5,0.5,2])
	at time 20 (objeto.pos = [50,0,0] ; objeto.scale = [1,1,1])
)
```

Dentro do método **animate on** você deve especificar **primeiro o número do frame que receberá o *keyframe*** e em seguida os **parâmetros que serão animados**.

Animação Recursiva ou Procedural

Vamos entender o significado das palavras Recursiva e Procedural:

Recursividade - termo usado para descrever o processo de repetição.
Procedural - tem a ver com sequência de processos.

Criar uma **Animação Recursivamente** ou **Proceduralmente** significa que você vai utilizar uma estrutura de repetição para fazer um determinado processo para animar diversos objetos (massas de objetos, dezenas, milhares ou até milhões objetos).

Exemplo em MaxScript:

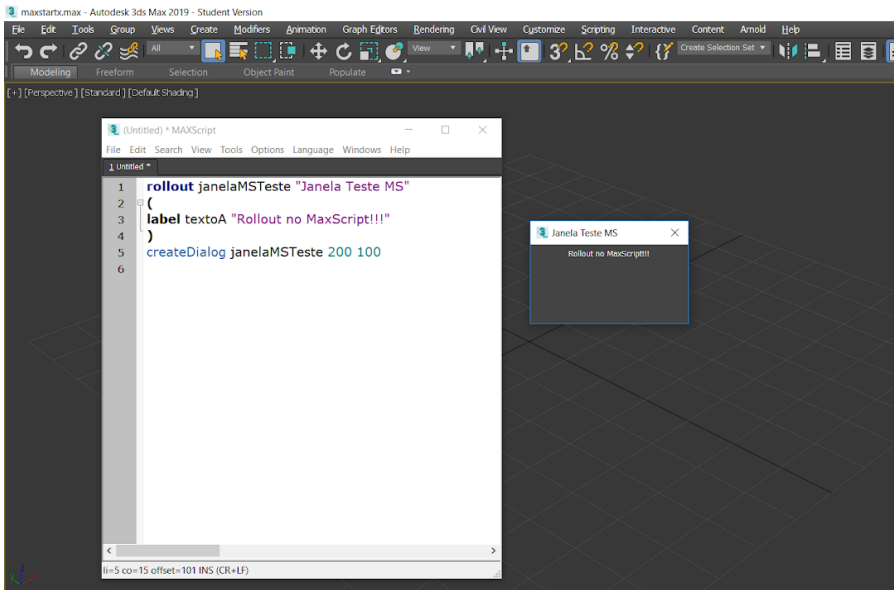
```
-- Vamos começar utilizando o exemplo Parte04 - 07_ArrayObjetos.ms que
você encontra no Pacote de Recursos de MaxScript indicados no início
deste documento.

delete $*
quantidade = 100
colunas = 5
largura = 5
altura = 3

for i = 0 to i = quantidade-1 do
(
	caixa = box width:largura length:altura height:10 pos:[(mod i
colunas)*largura,(floor(i/colunas))*altura,0] wirecolor:gray

	-- Com método animate dentro do laço, podemos manipular o
tempo das animações.
	-- Com o contador i, criando um atraso na animação do próximo
objeto.

	animate on (
		at time 0 (caixa.scale = [1,1,0.05])
		at time i (caixa.scale = [1,1,0.05])
		at time (i+9) (caixa.scale = [0.9,0.9,1.2])
		at time (i+10) (caixa.scale = [1,1,1])
	)

	completeredraw()
)

--Play na timeline.
playAnimation()
```

8. Criando Layouts e Interfaces com MaxScript

Janelas de Diálogo

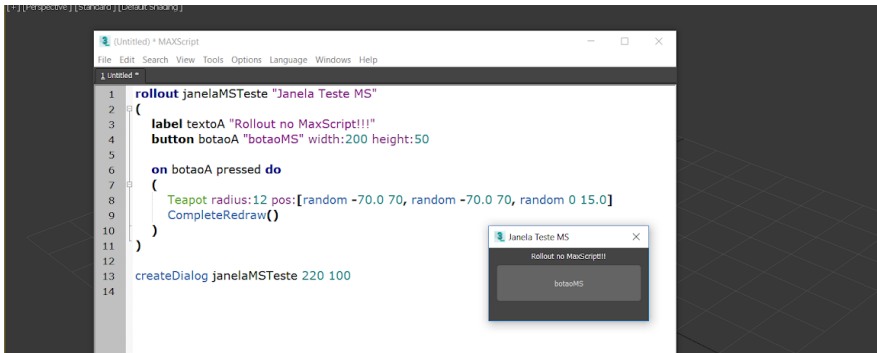
Podemos criar *layouts* e janelas e interface com MaxScript. Isto serve por exemplo para utilizar em *rigging* de personagens, para a exibição de caixas de diálogo, para criar ferramentas ou *plugins* ou inserir novas opções nos parâmetros de um objeto existente.

Rollouts

Os **Rollouts** são as Caixas de Diálogo. Podemos utilizar para criar *pop-ups*, alertas, exibir imagens ou mensagens, e até criar interfaces para *plugins* e ferramentas que criamos com MaxScript.

Exemplo em MaxScript:

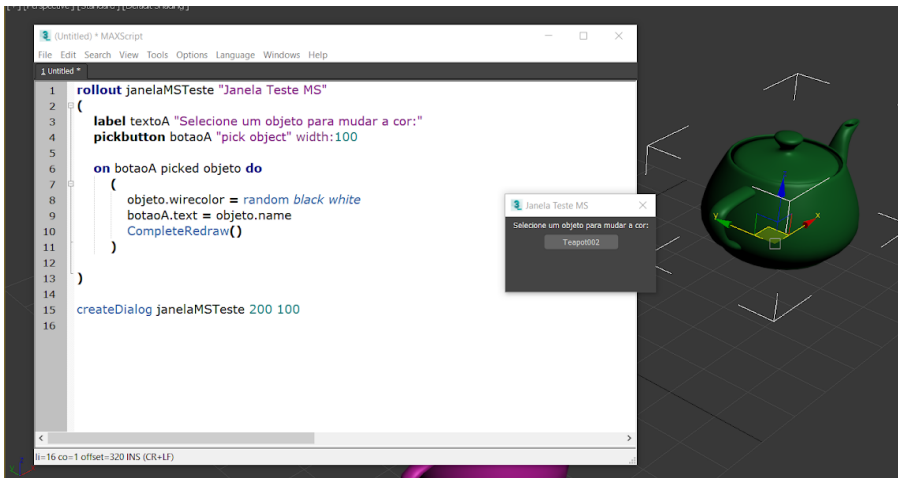
```
rollout janelaMSTeste "Janela Teste MS"
(
	label textoA "Rollout no MaxScript!!!"
)
createDialog janelaMSTeste 200 100
```

Button

O botão é o componente de interface que pode executar uma ação quando clicado. Além de declarar o Componente no *rollout* devemos declarar o método ***pressed*** para que a função seja executada.

Exemplo em MaxScript:

```
rollout janelaMSTeste "Janela Teste MS"
(
	label textoA "Rollout no MaxScript!!!"
	button botaoA "botaoMS" width:200 height:50

	on botaoA pressed do
	(
		Teapot radius:12 pos:[random -70.0 70, random -70.0 70,
random 0 15.0]
		CompleteRedraw()
	)
)

createDialog janelaMSTeste 220 100
```

Pick Button

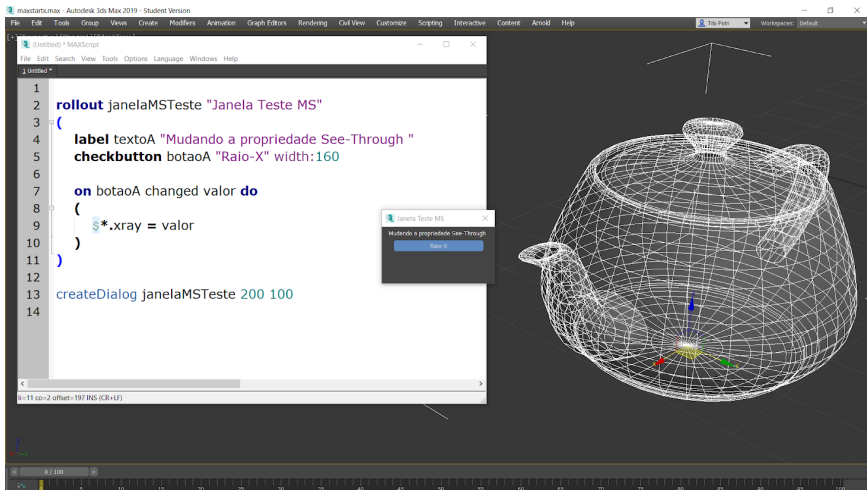
O **Pick Button** é o botão que ao ser clicado te permite selecionar um objeto na cena.

Perceba que o método ***picked*** nos retorna o objeto que foi clicado. Neste caso do exemplo, chamei ele de **objeto** mesmo.

Exemplo em MaxScript:

```
rollout janelaMSTeste "Janela Teste MS"
(
	label textoA "Selecione um objeto para mudar a cor:"
	pickbutton botaoA "pick object" width:100

	on botaoA picked objeto do
		(
			objeto.wirecolor = random black white
			botaoA.text = objeto.name
			CompleteRedraw()
```

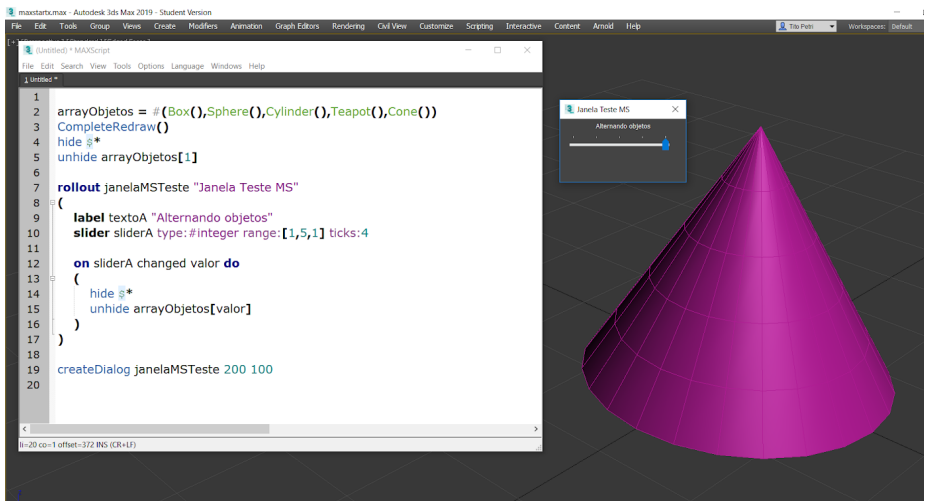
```
                        )

)

createDialog janelaMSTeste 200 100
```

Check Button

Este botão retorna ***true*** ou ***false*** quando pressionado, dependendo do seu estado. Serve para ligar ou desligar algum parâmetro. No exemplo acionaremos ou desativaremos a propriedade **See Through** (**.xray**), através do botão criado.

Exemplo em MaxScript:

```
rollout janelaMSTeste "Janela Teste MS"
(
	label textoA "Mudando a propriedade See-Through "
	checkbutton botaoA "Raio-X" width:160

	on botaoA changed valor do
	(
		$*.xray = valor
	)
)

createDialog janelaMSTeste 200 100
```

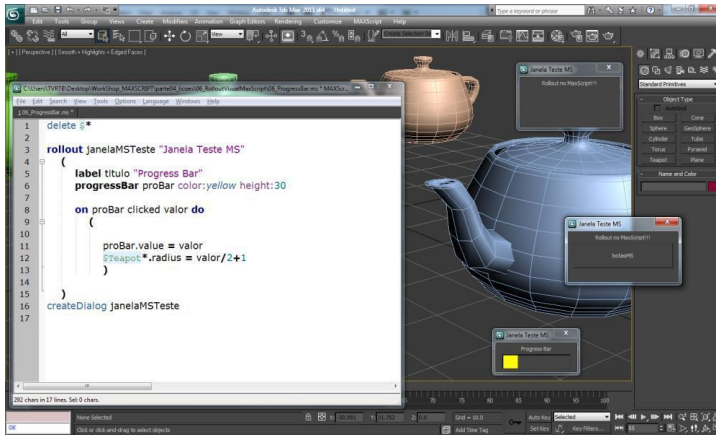
Slider

O ***Slider*** é o componente que nos retorna um intervalo entre dois valores, ao ser clicado ou movido. Este intervalo é especificado na criação do componente pelo parâmetro **range**.

Exemplo em MaxScript:

```
arrayObjetos = #(Box(),Sphere(),Cylinder(),Teapot(),Cone())
CompleteRedraw()
hide $*
unhide arrayObjetos[1]

rollout janelaMSTeste "Janela Teste MS"
(
		label textoA "Alternando objetos"
		slider sliderA type:#integer range:[1,5,1] ticks:4
```

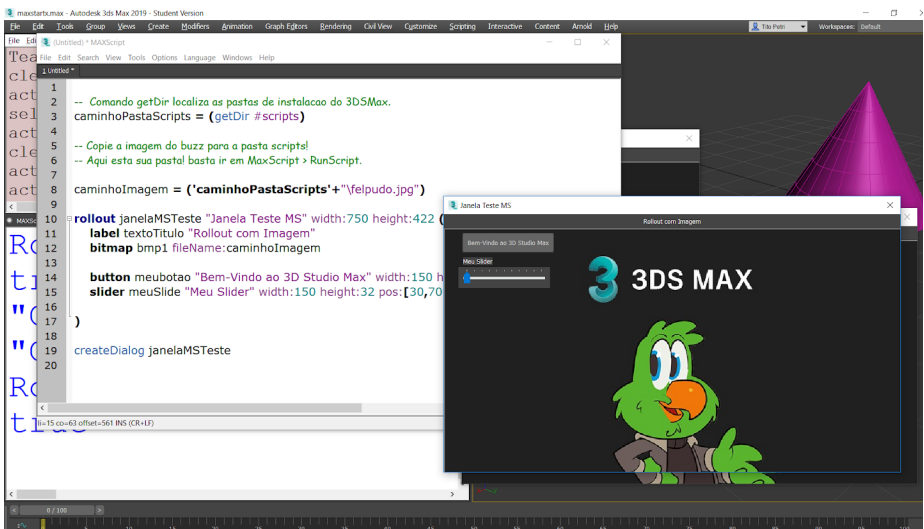
```
	on sliderA changed valor do
	(
		hide $*
		unhide arrayObjetos[valor]
	)
)

createDialog janelaMSTeste 200 100
```

Progress Bar

Similar ao slider, o **ProgressBar** também nos indica o intervalo entre dois valores. Geralmente, usamos este componente para indicar o progresso de um carregamento ou execução de algum processo.

Exemplo em MaxScript:

```
delete $*
bolinha = Sphere radius:1 segs:128

rollout janelaMSTeste "Janela Teste MS"
(

label titulo "Progress Bar"
progressBar proBar color:yellow height:30

		on proBar clicked valor do
		(
			proBar.value = valor
			bolinha.radius = valor/2+1
		)
)

createDialog janelaMSTeste
```

Exibindo um Bitmap

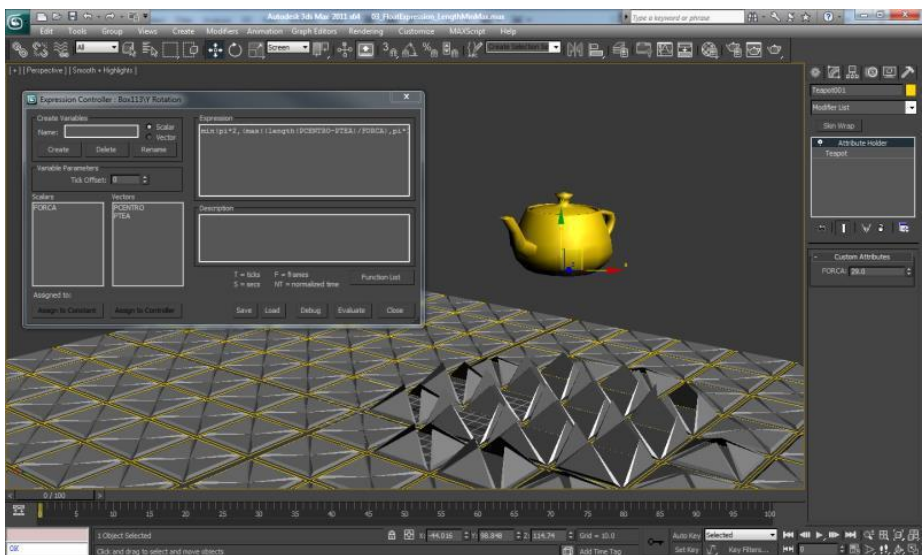
Podemos também carregar imagens nas nossas UIs (*User Interface*) ou *layouts.* Basta indicar o caminho (ou *url*) do seu Bitmap (arquivos **.bmp .jpg .png**, por exemplo).

Exemplo em MaxScript:

```
--  Comando getDir localiza as pastas de instalacao do 3DSMAX.
caminhoPastaScripts = (getDir #scripts)

-- Copie a imagem do buzz para a pasta scripts!
-- Aqui esta sua pasta! basta ir em MaxScript > RunScript.

caminhoImagem = ('caminhoPastaScripts'+"\felpudo.jpg")
```

```
rollout janelaMSTeste "Janela Teste MS" width:550 height:600 (
	label textoTitulo "Rollout com Imagem"
	bitmap bmp1 fileName:caminhoImagem
)

createDialog janelaMSTeste
```

Float Expression - Distância, Máximo e Mínimo

Na imagem a seguir, ilustro um exemplo que utilizei o MaxScript para animação. Nesta cena, ao aproximar ou afastar o *Teapot,* as tampas se abrem ou se fecham de acordo com a distância do objeto.

Para criar este efeito, utilizei o *constraint* **Float Expression** para trabalhar com expressões matemáticas na movimentação dos objetos.

No exemplo adiante, demonstro como adicionar o **Float Expression** utilizando apenas os códigos em MaxScript.

Exemplo em MaxScript:

```
objetoControle = Sphere segs:32 visibility:0.9 radius:25 wirecolor:red
material:(Standardmaterial diffuse:red selfillumamount:100)

quantidade = 200
colunas =15
largura = 10
altura = 10

for i = 0 to i = quantidade do
(
	objetoChamferBox = Cylinder radius:5 height:0.25 wirecolor:gray
pos:[(mod i
	colunas)*largura,(floor (i/colunas))*altura,0] wirecolor:yellow
	objetoParent = rectangle width:12 length:12 wirecolor:yellow
cornerradius:2
	pos:objetoChamferBox.pos steps:2
	objetoChamferBox.parent = objetoParent
	objetoControle.radius.controller = Bezier_Float()
	objetoFS = Float_Expression ()
	objetoFS.AddVectorTarget "posControlador"
objetoControle.pos.controller
	objetoFS.AddVectorTarget "posParent" objetoParent.pos.controller
	objetoFS.AddScalarTarget "falloff" objetoControle.radius.Controller
	expressaoFS =
"min(10,(max((length(posControlador-posParent)/falloff),0)))"
	objetoFS.SetExpression expressaoFS
	objetoChamferBox.radius.controller = objetoFS
)
```

Float Script – Seno e Coseno

Também com a ajuda do **Float Expression,** podemos utilizar as funções matemáticas **Seno** e **Cosseno** para gerar movimentos em círculo, ou em ondas.

Exemplo em MaxScript:

```
objetoControle = Sphere segs:32 visibility:0.9 radius:25 wirecolor:red
material:(Standardmaterial diffuse:red selfillumamount:100)

quantidade = 200
colunas =15
largura = 10
altura = 10

for i = 0 to i = quantidade do
(
```

```
		objetoChamferBox = Cylinder radius:5 height:0.25 wirecolor:gray 
pos:[(mod i
		colunas)*largura,(floor (i/colunas))*altura,0] wirecolor:yellow
		objetoParent = rectangle width:12 length:12 wirecolor:yellow 
cornerradius:2
		pos:objetoChamferBox.pos steps:2
		objetoChamferBox.parent = objetoParent
		objetoControle.radius.controller = Bezier_Float()
		objetoFS = Float_Expression ()
		objetoFS.AddVectorTarget "posControlador" 
objetoControle.pos.controller
		objetoFS.AddVectorTarget "posParent" objetoParent.pos.controller
		objetoFS.AddScalarTarget "falloff" objetoControle.radius.Controller
		expressaoFS = 
"min(10,(max((length(posControlador-posParent)/falloff),0)))"
		objetoFS.SetExpression expressaoFS
		objetoChamferBox.radius.controller = objetoFS
)
```
…

Parabéns querido aluno por chegar até aqui e ter adquirido mais este valioso conhecimento!

Se quiser aprender sempre mais sobre criação de Jogos e Aplicativos, não deixe de conhecer o Aprenda Programar, meu portal de cursos online onde você pode se especializar em:

- Algoritmos e Lógica de Programação
- Modelagem e Animação 3D
- Criação de Personagens para Jogos e Filmes
- Programação de Aplicativos Nativos para iOS e Android
- Criação de Games 2D, 3D e Realidade Virtual
- Realidade Aumentada e Visão Computacional
- Metodologia STEAM
- Robótica e Impressão 3D

Para virar aluno do Aprenda Programar você deve se inscrever pela plataforma Hotmart, no link abaixo.

Adquira seu acesso para sempre ao Aprenda Programar:
https://hotmart.com/product/en/aprenda-programar-com-tito-petri